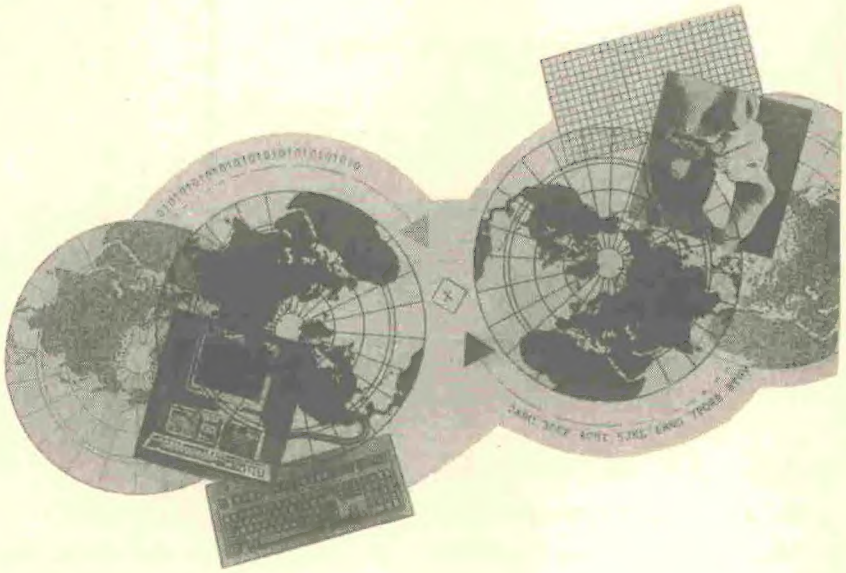
อธิการ มูลสาร

ปี ค.ศ. 2000 ในอีกมุมมอง



การจะแก้ไขปัญหาได้ดี ต้องค้นหาตัวปัญหาที่แท้จริงให้ได้เสียก่อน แล้วจึงวิเคราะห์ว่าปัญหาที่เกิดมีสาเหตุมาจากสิ่งใด จะแก้ปัญหาได้อย่างไร

ผู้เขียนได้รับฟังเรื่องราวของ ปี ค.ศ. 2000 มาจากหลายแหล่ง แต่สิ่งที่คนเหล่านั้นอดที่จะพาดพิงถึงไม่ได้คือเรื่องของวงการคอมพิวเตอร์ บ้างก็สันนิษฐานว่าเมื่อตัวเลขปีดังกล่าวถูกเก็บบนฐานข้อมูลเป็นศูนย์แล้ว การคำนวณเกี่ยวกับวันที่จะผิดพลาดหรือไม่ ลูกค้าของธนาคารที่ฝากเงินในปีนี้ (1997) ในอีก 3 ปีข้างหน้าจะไม่ได้รับดอกเบี้ยเลย ได้รับดอกเบี้ยรวม 97 ปี หรือต้องจ่ายดอกเบี้ยของ 97 ปีนั้นให้กับธนาคาร (ซึ่งคงจะต้องล้มละลายกันหมด) บ้างก็ว่าบริษัททเวนตี้-เซนจูรี่ฟ็อกซ์ จะได้รับผลกระทบจนต้องเปลี่ยนชื่อเป็นทเวนตี้วันเซนจูรี่ฟ็อกซ์ เพื่อยกระดับให้นำหน้าอยู่เสมอหรือไม่

บริษัทซอฟต์แวร์/ฮาร์ดแวร์ต่าง ๆ จะพากันตั้งชื่อผลิตภัณฑ์ของตนให้ลงท้ายด้วย 2000, 2K (1K เท่ากับ 1,000 ในความหมายของคนทั่วไป) หรืออะไรก็ได้ที่เกี่ยวกับตัวเลข 2000 ตัวอย่างเช่น Windows 2000 ซึ่งเป็นชื่อที่แก้ไม่น้อยทีเดียว ผู้เขียนว่าเรื่องนี้ให้เป็นหน้าที่ของผู้บริหารของแต่ละที่แต่ละแห่งเขาจัดการกันเองจะดีกว่า

ผู้เขียนมีแนวทางที่น่าสนใจในการแก้ไขปัญหาเรื่องของปี ค.ศ. 2000 นี้ ซึ่งเป็นเรื่องที่ทำได้ไม่ยากเลย ไม่ต้องแก้ไขแฟ้มข้อมูลแต่ประการใด เพราะการกระทำดังกล่าวนอกจากจะต้องขยายขนาดของฟิลด์ในแฟ้มข้อมูลแล้ว ยังเกิดความไม่แน่นอนขึ้นได้ระหว่างทำการ Convert ข้อมูลหากฐานข้อมูลที่ใช้ไม่มีระบบ Commitment Control ที่ดีพอ ข้อสำคัญฐานข้อมูลต้องใหญ่ขึ้นอีกแน่นอน

แจกแจงปัญหา

การจะแก้ไขปัญหาได้ดีเพียงใดนั้น ต้องค้นหาตัวปัญหาที่แท้จริงให้ได้เสียก่อน แล้วจึงวิเคราะห์ว่าปัญหาที่เกิดมีสาเหตุมาจากสิ่งใด และในขั้นตอนสุดท้าย จะแก้ปัญหาได้อย่างไร กระบวนการเหล่านี้เกิดขึ้นเองโดยไม่รู้ตัว แต่มักจะเรียงลำดับไม่ถูกต้อง ผู้แก้ปัญหาจำนวนมากมองไปที่หนทางแก้ปัญหาที่เกิดขึ้นแต่เพียงอย่างเดียว โดยไม่รู้จักตั้งคำถามให้กับตนเองทีละข้อแล้วหาคำตอบให้กับคำถามทุกข้อ ต่อไปนี้เป็นตัวอย่างของคำถามที่จะนำทางให้หาหนทางในการแก้ไขปัญหาได้ดีขึ้น

- ระบบปฏิบัติการที่ใช้ เก็บข้อมูลวันที่ไว้อย่างไร มีข้อจำกัดเพียงใด
- ฐานข้อมูลเก็บข้อมูลแบบวันที่ด้วยวิธีเดียวกับระบบปฏิบัติหรือไม่ ถ้าไม่ใช้วิธีใด

- โปรแกรมที่ใช้งาน ควบคุมการแสดงผลและการป้อนข้อมูลอย่างไร
- ผู้อ่านสามารถจัดการกับปัญหาได้เพียงใด แก้ไขโปรแกรมทั้งหมด หรือทำได้เพียงเปลี่ยนวิธีการรับข้อมูลและแสดงผล
- จะเลือกใช้วิธีใดข้างต้น
- หากจำเป็นต้องเปลี่ยนแปลงแฟ้มข้อมูล สามารถกระทำได้หรือไม่ เพียงใด

ข้อมูลชนิดวันที่

ระบบปฏิบัติการดอส เก็บข้อมูลชนิดวันที่ไว้เป็นแบบพิเศษ ผู้ใช้ไม่จำเป็นต้องทราบว่าข้อมูลถูกจัดเก็บอย่างไร แต่ต้องทราบลำดับการแสดงผลและวิธีในการป้อนข้อมูล เช่น ปี-เดือน-วัน, วัน-เดือน-ปี หรือ เดือน-วัน-ปี ซึ่งแบบหลังจะพบเห็นได้บ่อยที่สุด (mm-dd-yy) เพราะเป็นรูปแบบพื้นฐานของดอส (ถ้าใช้วินโดวส์สามารถเปลี่ยนได้จาก Control Panel) โดยอักษรแต่ละตัวจะแทนตัวเลขที่ผู้ใช้ต้องป้อน อักษรคั่นมีไว้เพื่อแยกวัน/เดือน/ปี ออกจากกัน บางครั้งอักษรคั่นอาจต่างไปจากนี้ แต่สิ่งที่สำคัญอยู่ที่ผู้อ่านไม่สามารถป้อนวันที่แบบผิด ๆ เช่น 13-01-97 หรือ 31-13-86 ได้ เพราะระบบจะตรวจสอบให้เองอัตโนมัติ

เราจะมุ่งความสนใจไปที่ส่วนของตัวเลขแสดงปี หากเราป้อนวันที่เป็น 01-01-00 (ซึ่งน่าจะหมายถึงวันที่ 1 ม.ค. ค.ศ. 2000) ดอสจะเก็บปีที่เราป้อนอย่างไร คำตอบก็คือดอสจะแสดงข้อความผิดพลาด "Invalid date" ออกมาแทน แล้วเราจะป้อนปี 2000 ได้อย่างไร

โชคดีที่ถึงแม้คำสั่ง Date ของดอสจะแสดงคำอธิบายการป้อนวันที่เป็น mm-dd-yy แต่เราสามารถป้อนปีเป็นตัวเลข 4 หลักได้ เช่น 01-01-2000 แน่นอนว่าคราวนี้ดอสยอมรับและเก็บวันที่ดังกล่าวเป็นวันที่ของระบบ ผู้เขียนจึงแนะนำให้ฝึกป้อนวันที่โดยคีย์ปีเป็นตัวเลข 4 หลักเสียตั้งแต่บัดนี้ (มีเวลาให้ฝึกแค่ 3 ปีเท่านั้น) อย่างไรก็ตาม ตอนนี้เราได้คำตอบแล้วว่าดอสไม่ได้จำกัดวันที่อยู่แค่ปี 1999 เท่านั้น

เมื่อทดลองป้อนปีที่มากขึ้น ก็พบว่าไปสิ้นสุดที่ปี 2099 ผู้เขียนไม่สามารถป้อนปี 2100 ได้ แน่นอนว่าบนวินโดวส์ก็ได้รับผลกระทบนี้ แต่ก็ทำให้วางใจได้ว่า เรามีเวลาเหลืออยู่อีกถึง 99 ปีนับจากปี 2000 และถ้าตอนนั้นดอสยังอยู่ล่ะก็ ผู้พัฒนาคงจะไม่มองข้ามข้อนี้ไปหรอก ความพยายามในการทดสอบทั้งหมดนี้ดูได้จากลิสติงที่ 1

```
C:\>DATE
Current date is Sat 01-01-2000
Enter new date (mm-dd-yy): 13-01-97

Invalid date
Enter new date (mm-dd-yy): 31-13-86

Invalid date
Enter new date (mm-dd-yy): 01-01-00

Invalid date
Enter new date (mm-dd-yy): 01-01-2000

C:\>DATE
Current date is Sat 01-01-2000
Enter new date (mm-dd-yy): 01-01-2099

C:\>DATE
Current date is Thu 01-01-2099
Enter new date (mm-dd-yy): 01-01-2100

Invalid date
Enter new date (mm-dd-yy):
```

ลิสติงที่ 1

ฐานข้อมูลกับวันที่

ระบบฐานข้อมูลส่วนใหญ่จะเก็บวันที่ในรูปแบบวันที่ (เอ๊ะ! พูดยังไง) ฉะนั้นผู้ใช้จึงหายห่วงในเรื่องนี้ได้ (อย่างน้อย 99 ปี) แต่ถ้าฐานข้อมูลเก็บรูปแบบอื่น เช่น ตัวอักษรขนาด 6-8 ไบต์ หรือ Long Integer ฯลฯ ซึ่งผู้เขียนจะได้กล่าวถึงการจัดการต่อไป

ปัญหาของฟิลด์แบบวันที่ในขั้นตอนนี้จึงเหลือเพียงโปรแกรมที่ใช้อยู่ บังคับให้ผู้ใช้คีย์ตัวเลขปีได้เพียง 2 หลักเท่านั้นหรือไม่ ถ้าเป็น Microsoft Access การแก้ไขก็เพียงไปขยายช่องวันที่ให้กว้างขึ้นและเปลี่ยน Masked ที่ใช้ควบคุมการป้อนข้อมูลและแสดงผลให้เป็น 4 หลัก แต่ถ้าเป็น Visual Basic, Delphi หรือ Tool อื่น ๆ นอกจากจะต้องแก้ไขข้างต้นแล้ว อาจต้องคอมพายล์โปรแกรมใหม่ทั้งหมดด้วย แต่ก็คุ้มที่จะทำ

เรื่องของแฟล็ก

ก่อนที่จะพูดถึงแนวทางในการแก้ปัญหาต่อ ผู้เขียนมีประสบการณ์กับฟิลด์วันที่บนระบบ AS/400 ของไอบีเอ็ม ฟิลด์ประเภทดังกล่าวถูกเก็บอยู่ในรูปตัวแปร Zoned ขนาด 7 Digit ซึ่งผู้อ่านคงสงสัยว่าตัวเลขหลักที่ 7 นั้นเกินความจำเป็นหรือไม่ แต่จริง ๆ แล้วมันมีประโยชน์และจำเป็นอย่างมากทีเดียว

ข้อมูลวันที่จะถูกเรียงลำดับในรูปแบบ fyymmdd ซึ่งนอกจากวิธีเรียง yymmdd จะให้ข้อดีในการมองไปที่ข้อมูลได้ง่ายแล้ว (ไม่ต้องสับสนว่าอะไรมาก่อน-หลัง) ยังมีส่วนที่เรียกว่าแฟล็ก (flag) ที่เผื่อเหลือเผื่อขาดไว้ให้โปรแกรมเมอร์ หมายความว่า หากปีเกิดล้นขึ้นมาเมื่อไหร่ล่ะก็ ให้ทดข้ามไปที่แฟล็กดังกล่าว เสียก็หมดปัญหา เช่น ทดจาก 99 เป็น 1000 เมื่อสิ้นปี 1999 ก็จะทำให้สามารถเก็บข้อมูลเพิ่มขึ้นได้ถึง 900 ปีเลยทีเดียว (แฟล็กเป็น 1-9)

มีประเภทข้อมูลดี ๆ แบบนี้ทั้งที แต่ทำไมโปรแกรมเมอร์ส่วนใหญ่ไม่มองเห็นความสำคัญตรงจุดนี้ก็ไม่ทราบ ผู้เขียนพบโปรแกรมเมอร์จำนวนมากที่นอกจากจะไม่ใช้แฟล็กดังกล่าวแล้ว ยังเอา 0 ใส่เข้าไปเพิ่มงานให้ซีพียูเล่นอีกต่างหาก นัยว่าจะเพิ่มจำนวน Line ของโปรแกรมเพื่อให้ดูเป็นมืออาชีพที่

เขียนโปรแกรมได้ยาวเหยียดหรือกลัวแฟล็กดังกล่าวจะไม่เป็นศูนย์ชัวร์ ๆ ก็ไม่ทราบได้ ฝากเรื่องนี้ให้โปรแกรมเมอร์ AS/400 คิดสักนิด

สำหรับโปรแกรมเมอร์บน PC ถ้าจะออกแบบฐานข้อมูลล่ะก็ ตัวเลข 7 หลักแบบนี้น่าสนใจไม่น้อยเลยทีเดียว (ตัวแปรขนาด Long Integer ก็ใช้ได้แล้ว)

ตัวอักษรขนาด 8 ไบต์

ถ้าโปรแกรมที่ใช้งานอยู่เก็บข้อมูลลงบนฐานข้อมูลเป็นอักษร 6 ไบต์ (ไม่ทราบว่าจะประหยัดอะไรกันนัก) ทางแก้ค่อนข้างยากนิดหน่อย แต่ถ้าเป็นขนาด 8 ไบต์ล่ะก็ มีวิธีอีกอีกอื้น (แปลว่ามหาศาล มโหฬาร ล้านแปดข้อ) ในการแก้ไข แต่ต้องทำใจกันหน่อยล่ะว่า ต้องแก้โปรแกรมแล้วคอมพายล์กันใหม่แน่นอน

ผู้เขียนจะพูดถึงการแก้ปัญหาของฟิลด์วันที่ที่เก็บเป็นตัวอักษรขนาด 8 ไบต์ก่อน (mm-dd-yy หรือ dd-mm-yy) ถ้าลองคิดตามผู้อ่านจะเห็นว่าตัวเลขที่จะเก็บมีเพียง 6 ตัวเท่านั้น หมายความว่าตัวอักษรที่เป็นตัวคั่น เช่น "-" หรือ "/" ถูกเก็บลงไปด้วยถึง 2 ตำแหน่ง (1 ใน 4 ของฟิลด์)

สมมติว่าในฐานข้อมูลเก็บวันที่รูปแบบ dd-mm-yy ลองนึกถึงตัวอย่างของแฟล็กที่ผ่านมา เราสามารถแทนที่เครื่องหมาย "-" ตัวที่ติดอยู่กับปี (ตำแหน่งที่ 6 นับจากซ้ายมือ) โดยถ้าตำแหน่งดังกล่าวเป็น "-" หรือ "0" น่าจะหมายถึงปี 19yy แต่ถ้ามีค่ามากกว่า "0" ให้ใช้สูตรดังนี้

```
String(Val(Mid$(strDate, 6, 1)) * 100)
+ 1900) + Val(Right$(strDate, 2)))
```

สูตรนี้อาจจะยาวไปสักนิด ผู้อ่านอาจใช้ฟังก์ชันในลิสติงที่ 2 แทนได้ ซึ่งจะเข้าใจได้ง่ายกว่า ในกรณีตรงกันข้ามเราสามารถใช้วิธีกลับกันเพื่อแปลงค่ากลับเป็นตัวอักษร 3 ตัว เพื่อบันทึกลงในฐานข้อมูล ฟังก์ชัน DateY3() ในลิสติงที่ 2 จะแปลงปีเป็นอักษร 3 ตัวเพื่อวัตถุประสงค์ดังกล่าว ส่วนฟังก์ชัน DateY4() ในลิสติงเดียวกันจะแปลงกลับเป็นปี ค.ศ. ที่ผู้ใช้ทั่วไปเข้าใจได้

```
'Formats from "dd-mmfyy" to "dd-mm-yyyy"
Function DateY4(d As String) As String
   Dim yy As Integer
   Dim flag As Integer

   yy = Val(Right(d, 2))
   flag = Val(Mid(d, 6, 1))

   DateY4 = String(1900 + (flag*100) + yy)
End Function

'Formats from "dd-mm-yyyy" to "dd-mmfyy"
Function DateY3(d As String) As String
   Dim yyyy As Integer
   Dim flag As Integer
   Dim f As String

   yyyy = Val(Right$(d, 4))
   flag = IIF(yyyy<2000,0,(yyyy-1900)/100)
   f = String(flag)

   DateY3 = Left$(d, 4) & f & Right$(d, 2)
End Function
```

ลิสติงที่ 2

ด้วยจำนวนปีที่เพิ่มขึ้นมา 900 ปีนี้ ถ้ายังไม่เพียงพอกับความต้องการ ก็ยังเหลืออักษรคั่นอีกตัวในตำแหน่งที่ 3 ให้ใช้ แต่คราวนี้ผู้ใช้จะได้จำนวนปีเพิ่มขึ้นถึง 9,000 ปีเลยทีเดียว (ลองคิดดูว่าทำไม)

ตัวอักษรขนาด 6 ไบต์

ถึงคราวของฟิลด์วันที่ขนาด 6 ตัวอักษรหรือ 6 ไบต์ เราเจอปัญหาใหญ่ตรงที่อักษรทั้ง 6 ตัวถูกใช้ไปหมดแล้ว และไม่มีที่ให้ขยับขยายอีกด้วย ทำอย่างไรจึงจะแก้ปัญหานี้ได้

สมมติว่าข้อมูลเก็บในรูปแบบ ddmmyy (6 ไบต์พอดี) ด้วยรูปแบบนี้ มองยังไงก็ใช้กับเทคนิคแฟล็กไม่ได้ แต่ยังพอมีวิธี ใช้วิธีการเดียวกับเลขฐาน 16 นั่นเอง กล่าวคือ เมื่อใดก็ตามที่ตัวเลขหลักหน้าของปีมีค่ามากกว่า 9 ให้แทนด้วยอักษร A, B, C, ... Z ตามลำดับ (คล้ายกับการแทนที่ตัวเลขด้วยตัวอักษรของระบบเลขฐาน 16 - ขอเรียกวิธีนี้ว่าการแปรค่า) ด้วยวิธีนี้เราจะได้จำนวนปีเพิ่มขึ้นถึง 2,600 ปี (A-Z) และหากยังไม่พอใจ เรายังสามารถใช้ความแตกต่างของตัวอักษร คือใช้อักษรตัวพิมพ์เล็กเข้ามาร่วมด้วย ก็จะได้จำนวนปีเพิ่มอีก 2,600 ปี (a-z) ซึ่งถ้าใครจะใช้ตัวอักษรพิเศษมาประยุกต์ใช้ ก็สามารถทำได้อีกสารพัดวิธี ไม่ว่ากัน

ด้วยวิธีการนี้ ข้อมูลที่ได้อาจจะดูขัดตาไปบ้าง เช่น 0112A2 ซึ่งหมายถึงวันที่ 1 ธันวาคม ค.ศ. 2002 แต่การแสดงผลถูกควบคุมโดยโปรแกรม (คุมมาตั้งแต่แรกแล้ว) ฉะนั้นผู้ใช้จะไม่รู้สึกถึงการเปลี่ยนแปลงได้เลย

ฟังก์ชันที่ทำหน้าที่แปลงไป-กลับ มีอยู่ในลิสติงที่ 3 โดยฟังก์ชัน DWrite() จะแปลงวันที่รูปแบบ dd-mm-yyyy ให้เหลือเพียง ddmmxy (x จะแปรค่าได้ดังที่กล่าวมาแล้ว) เพื่อบันทึกในฐานข้อมูล ส่วนฟังก์ชัน DRead จะอ่านฟิลด์วันที่ขนาด 6 ตัวอักษรข้างต้น แล้วแปลงกลับเป็นวันที่ขนาด 8 ตัวอักษร (dd-mm-yyyy) ที่ประกอบด้วยปี 4 หลักที่เราต้องการ

ผู้เขียนไม่แนะนำให้ใช้วิธีการแปรค่านี้กับตัวอักษรตัวหลังของปี เพราะจะทำให้สับสนและควบคุมยาก แต่ถ้าขยันล่ะก็ เชิญ!!!

ลิสติงที่ 3

```
'Formats from "dd-mm-yyyy" to "ddmmxy"
Function DWrite(d As String) As String
   Dim yyyy As Integer
   Dim flag As Integer
   Dim f As String

   yyyy = Val(Right$(d, 4))
   flag = IIF(yyyy<2000,0,(yyyy-1900)/100)
   If flag < 10 Then
      f = String(flag)
   Else
      f = Chr$((flag-10) + Asc("A"))
   End If

   Dwrite = Left$(d, 2) & Mid$(d, 4, 2) &
      f & Right$(d, 1)
End Function
```

```
'Formats from "ddmmxy" to "dd-mm-yyyy"
Function DRead(d As String) As String
   Dim y As Integer
   Dim flag As String
   Dim yyyy As Integer

   y = Val(Right$(d, 1))
   flag = Mid$(d, 5, 1)
   If flag < "A" Then
      f = Val(flag)
   Else
      f = Asc(flag)-Asc("A")+10
   End If
   yyyy = 1900 + (f * 100) + y

   Dread = Left$(d, 2) & "-" &
      Mid$(d, 3, 2) & "-" & String(yyyy)
End Function
```

Long Integer

ถ้าโปรแกรมเก็บข้อมูลในรูปแบบ Long Integer อยู่แล้ว การแก้ไขจะค่อนข้างง่าย ผู้เขียนแนะนำให้เก็บในรูปแบบ fyymmdd ดังที่กล่าวมาแล้วจะเป็นการดียิ่ง ดูรายละเอียดในหัวข้อ "เรื่องของแฟล็ก" ส่วนวิธีการจัดการนั้นให้ใช้วิธีเดิม แต่ถ้าเก็บอยู่ในรูปแบบอื่น เช่น fmmddyy ผู้เขียนยกให้เป็นหน้าที่ของผู้อ่านในการคิดวิธีแก้ปัญหาก็แล้วกัน ไม่ยากหรอก

จบแล้ว

การแก้ปัญหาของผู้เขียน ใช้วิธีการแก้ไขโปรแกรม เนื่องจากว่าการเปลี่ยนแปลงฐานข้อมูล (หรือตัวข้อมูล) ถึงจะทำได้ไม่ยากนัก แต่เป็นวิธีที่ไม่ฉลาดนักและควรทำเป็นขั้นตอนสุดท้ายเมื่อหมดทางเลือกแล้วเท่านั้น อย่างไรก็ตาม อย่าตระหนกกับตัวเลขนัก ปัญหาทุกอย่างมีหนทางแก้ไขเสมอ ปี ค.ศ. 2000 ก็เป็นเพียงตัวเลขเช่นกัน สวัสดี

เวบไซท์เพจเจอร์

ในปัจจุบันนอกจากจะส่งเพจ-เจอร์โดยวิธีกดเบอร์โทรศัพท์ไปยังผู้ให้บริการ และฝากข้อความกับโอเปอร์เรเตอร์เสียงใสแล้ว คุณสามารถส่งข้อความไปยังเพจเจอร์ผ่านทางอินเทอร์เน็ต ซึ่งผู้ให้บริการต่างก็ได้เปิดให้บริการผ่านเวบไซท์ของตนเองดังนี้

โฟนลิ้งค์ 151/152

http://www.phonelink.net

อิสซี่คอล 1500

http://www.easycall.lenso.co.th

โพสต์เทล 1188

http://www.samart.co.th/pager

เวิล์ดเพจ 141/142

http://www.wnet.net.th

ฮัทจิสันเพจโฟน 161/162

http://www.hutchison.co.th

แพคลิ้งค์ 1144
(อยู่ในช่วงดำเนินการจัดทำโฮมเพจ)